Sentiment Analysis

A Probabilistic Approach

S. A. Gieske S. Laan D. S. Ten Velthuis
C. R. Verschoor A. J. Wiggers

Faculty of Science (FNWI)
University of Amsterdam

February 3, 2012

Outline

Outline

The goal of the project

Project Description

Performing sentiment analysis on messages about the EO

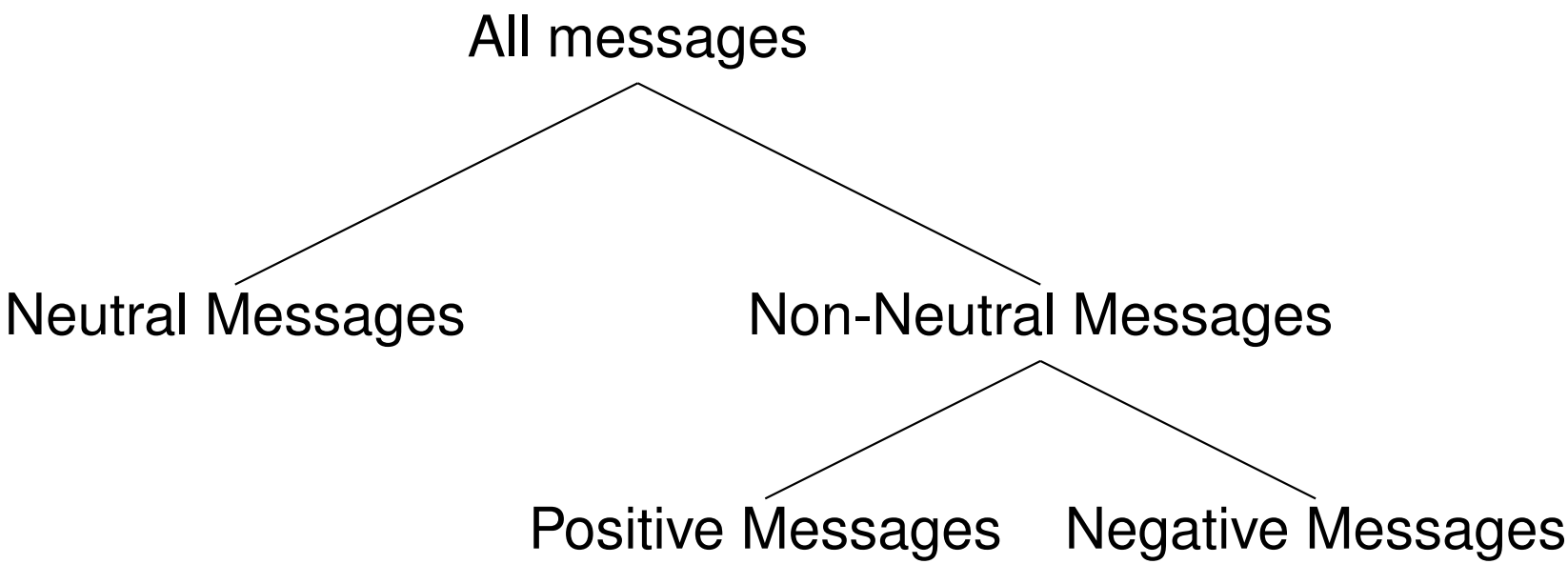
- Classification Sentiment vs. Non Sentiment
- Classification Positive vs. Negative

Outline

Approach

- Preprocessing of the data
- Perform machine learning algorithms on data
- Use best algorithms to classify real time on server

Hierarchical Classification



Outline

Outline

Dataset Analysis

Dataset messages EO

10.000 messages, 19 features per message

Ony 3 features used:

- Source
- Sentiment
- Message contents

Data Cleaning

- Shorten words, e.g. ‘saaaaaaai’ to ‘saaai’
- Stemmer

Data Reduction

- Only use Twitter messages (83% of all messages)
- Remove articles, personal pronouns and prepositions
- Substitute smileys with words
- Remove some punctuation marks (not ! ?)

Outline

Outline

Classification

Weighted Sum Probability

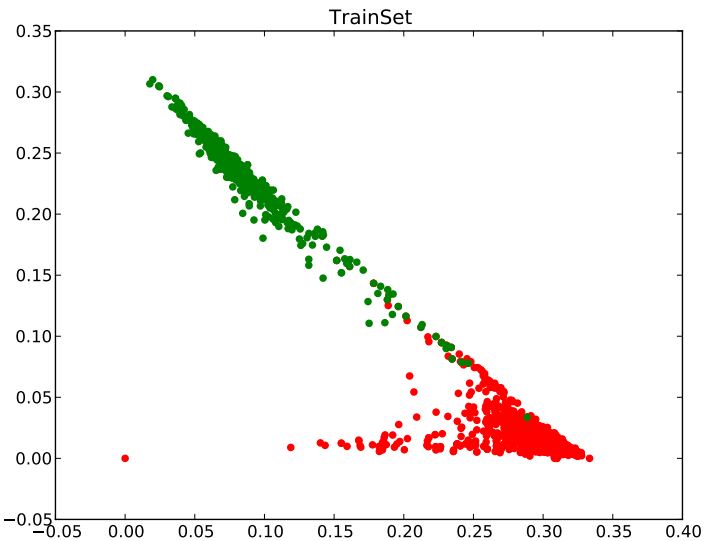
- Extract features
- Assign sentiment probabilities to features

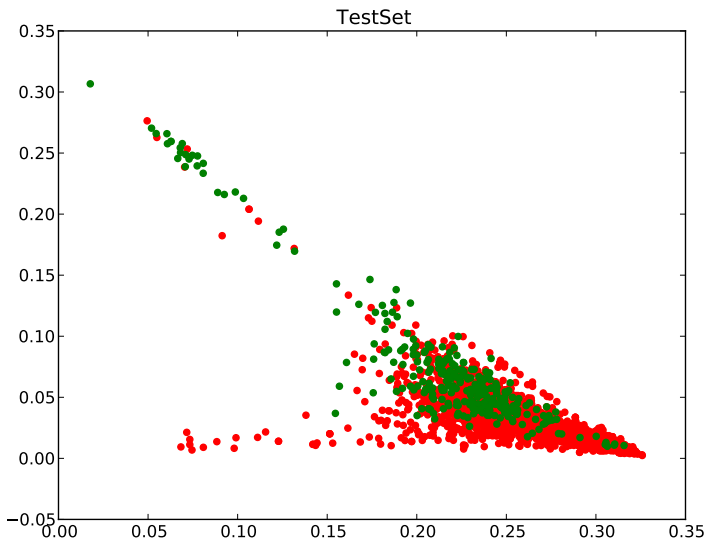
$$P(feature) = \frac{\sum feature \in C_1}{\sum feature \in C_1 \cup C_2} \quad (1)$$

- Assign sentiment probabilities to sentences

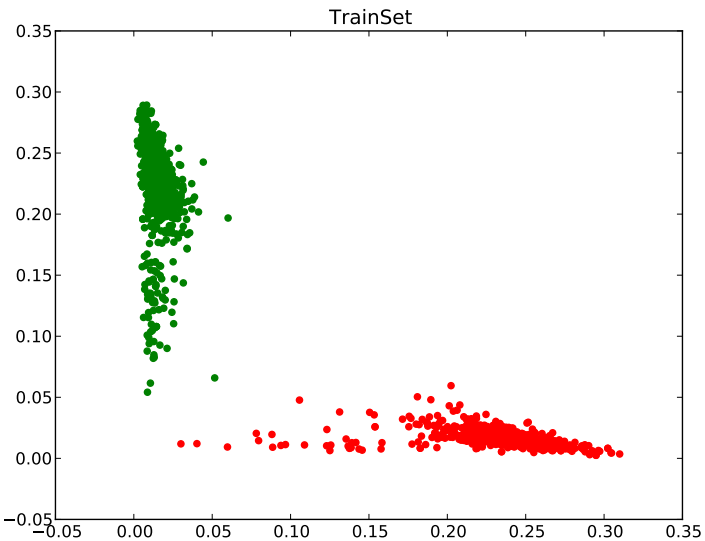
$$P(s) = \frac{1}{n} \sum_{f \in s} P(f) \quad (2)$$

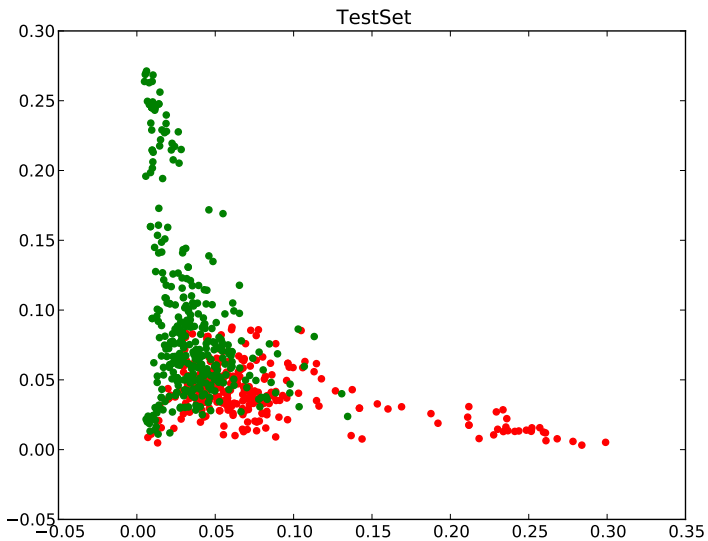
WSP: Neutral vs Non-Neutral





WSP:Positive vs Negative





Perceptron

Algorithm

Train linear treshold

Input : Sentence probabilities, sentence values

Output : Treshold

Results & Conclusion

Results : High precision OR recall, never both

Conclusion : Linear threshold not good enough

Support Vector Machine

Algorithm

Fit in featurespace that binds features to classes

Input : Features in vector

Output : Number belonging to class

Results & Conclusion

Results : Not very good recall/precision/accuracy

Conclusion : Fit can not be made on these features
or more data needed to find clear boundary

Naive Bayes

Algorithm

Prior and likelihood lead to posterior

Input : Features from sentence

Output : Probability

Results & Conclusion

		Recall	Accuracy	Precision
Results :	Positive	0.43	0.81	0.39
	Negative	0.40	0.71	0.17
	Neutral	0.61	0.59	0.79

Conclusion : Low recall and precision

Multiclassification with Perceptron

Algorithm

Specialized perceptron for each class, one vs. all

Input : Sentence probability for class

Output : Most likely class

Results & Conclusion

Results :

Test \Real	True	False
True	41	22
False	113	418

Accuracy = 0.77

Precision = 0.65

Time taken: 33 sec

Conclusion : Moderate results

Entropy

Algorithm

Words with highest likelihood for class

Input : Corpus

Output : Most likely class

Results & Conclusion

		Recall	Accuracy	Precision
Results :	Positive	0.42	0.74	0.25
	Negative	0.22	0.88	0.47
	Neutral	0.76	0.69	0.80

Conclusion : No satisfying results for positive and negative classification

Neural Network

Algorithm

Backpropagation

Input : Features from sentence

Output : Value for outputnodes (classes)

Results & Conclusion

Results : Training time = 2.85 hours for 500 sentences.

Test \Real	True	False
True	14	6
False	40	94

Conclusion : Still many messages with sentiment incorrectly classified.
Possible cause: ratio of messages with sentiment and nonsentiment.

Outline

Webserver Framework

Request → Server (PHP/PYTHON) → Result (XML)

Request http://url.com/?**dataset**=1&**message**=De EO is cool!

Result XML File (Containing: Status, Message, Sentiment, Accuracy, Precision, Recall)

Demo

Action...

Outline

Conclusion

- All learning algorithms have their (dis)advantages
- Multiclassification with perceptron and Neural Networks give best results
- Not enough data

Questions?